

A DMAP PROGRAM FOR THE SELECTION OF ACCELEROMETER LOCATIONS IN MSC/NASTRAN

Jeff Peck and Isaias Torres
NASA/Marshall Space Flight Center
Huntsville Alabama

Abstract

A new program for selecting sensor locations has been written in the DMAP (Direct Matrix Abstraction Program) language of MSC/NASTRAN. The program implements the method of Effective Independence for selecting sensor locations, and is executed within a single NASTRAN analysis as a "rigid format alter" to the normal modes solution sequence (SOL 103). The user of the program is able to choose among various analysis options using Case Control and Bulk Data entries. Algorithms tailored for the placement of both uni-axial and tri-axial accelerometers are available, as well as several options for including the model's mass distribution into the calculations. Target modes for the Effective Independence analysis are selected from the MSC/NASTRAN ASET modes calculated by the "SOL 103" solution sequence. The initial candidate sensor set is also under user control, and is selected from the ASET degrees of freedom. Analysis results are printed to the MSC/NASTRAN output file (*.f06), and may include the current candidate sensors set, and their associated Effective Independence distribution, at user specified iteration intervals. At the conclusion of the analysis, the model is reduced to the final sensor set, and frequencies and orthogonality checks are printed. Example results are given for a pre-test analysis of NASA's five-segment solid rocket booster modal test.

Introduction

For small, simple structures, accelerometer locations can be selected based on geometry, experience and intuition with good results. However, as structures become larger and more complex, identification of good locations for measurements is less intuitive. There are many methods and tools available to assist the engineer in selection of "good" or "optimal" measurement locations. However, many of these require the engineer to perform multiple finite

element analyses, or to export model data to external programs. These processes are often confusing and contain many opportunities for error. For example, using MSC/NASTRAN alone, a set of sensor location may be calculated using modal kinetic energy. These sensor locations may then be used to form a reduced model, and generate a cross-orthogonality matrix to gauge the accuracy of the reduction. Performing this analysis requires three basic steps; 1) run an analysis to compute full model modes and KE, saving mode shapes into punch file, 2) form ASET entries from KE results, and 3) run analysis with "premaca" alter, including ASET entries and punch file with modes.

A simpler and less error prone process would be for the finite element software to perform a selection process within a single analysis. To this end, a DMAP (Direct Matrix Abstraction Program) program has been written for MSC/NASTRAN which implements the Effective Independence technique (Efi) developed by D. C. Kammer^{1,2}. The DMAP program is executed as a single analysis within the normal modes solution sequence (SOL 103) and analysis options are controlled with a set of case control and bulk data entries. Results are printed in the NASTRAN "f06" file, and may consist of a DOF set for the final sensors, orthogonality checks and ASET entries for use in subsequent analyses.

Effective Independence

The effective independence techniques for selecting sensor location for modal tests were developed from the standpoint of the structural dynamist. From this standpoint, the best locations for the sensors are those locations that allow the best correlation and validation of a finite element model. To this end the independence of the test mode shapes is of prime importance. The effective independence techniques are iterative in nature, and within each iteration a vector is developed that measures the relative contribution of each candidate sensor to the

with the square root of the partitioned diagonal mass matrix. Thus, form $\underline{m}^{1/2}$ by extracting the diagonal of the mass matrix M , taking the square root of each term on the diagonal, and partitioning down to the DOFs contained in φ . Then in step two of each of the algorithms above, replace φ with $\underline{m}^{1/2}\varphi$, and the algorithm then proceeds as normal. Subsequent iterations do not need to reform the matrix $\underline{m}^{1/2}\varphi$, since simply partitioning $\underline{m}^{1/2}\varphi$, will have the same result.

When mass weighting with Guyan reduction is to be used, the situation is different. Here, the matrix $\underline{m}^{1/2}\varphi$ must be re-calculated within each iteration. This is because as DOFs are removed from the candidate sensor set, and the mass from the removed DOF is redistributed to the remaining DOFs according to the Guyan transformation matrix, the diagonal elements of the reduced mass matrix will change. Thus $\underline{m}^{1/2}$ will be different from one iteration to the next. However, once the reduced mass matrix is formed, the procedure of forming $\underline{m}^{1/2}\varphi$ is the same as in the partitioning case.

Efi DMAP Program

The EFI program was written as a MSC/NASTRAN DMAP program for several reasons. First, since most of the FEM models used by the dynamics group at MSFC are NASTRAN models, using a DMAP program means that no models or data need to be exported to some other program. Second, a NASTRAN DMAP program can be controlled with a set of custom NASTRAN entries (i.e. PARAM, DTI, etc...), this is a small change to a familiar interface for most dynamists, so learning to use the program should not be very difficult. Third, a DMAP program is able to take advantage of the highly optimized modules in NASTRAN for performing such operations as eigenvalue analysis and matrix products. And fourth, a properly written DMAP program should be able to run on any platform for which NASTRAN is available; therefore, compiler and porting issues are not a concern.

The DMAP program that resulted from this effort did not quite live up to all of the points mentioned in the previous paragraph. The most notable exception is related to the performance of the tri-axial routines. In general, DMAP programs run most efficiently when the majority of the computations are done within a module (subroutine). For the tri-axial routine, the calculation in step four required a loop to perform the triple product and determinant calculation for each

grid. This loop repeatedly called DMAP modules to operate on small pieces of matrices. Repeatedly calling these modules added a significant amount of "call overhead" to the execution of the program, and can seriously degrade the overall performance when the candidate sensor set is large. This is in contrast to the uni-axial algorithm that can be performed without any inner loops over the individual DOFs. To overcome this issue, the ISHELL module was used to export the inner loop calculations of the tri-axial algorithm to an external FORTRAN program. The FORTRAN program is much more efficient at performing the repetitive, small matrix calculations than the corresponding loop in DMAP. As a result, the DMAP program may be executed as a stand-alone DMAP program, or as a combination of DMAP and FORTRAN code. The DMAP portion of the program is written to be executed within the normal modes solution sequence (SOL 103), and assumes execution control immediately after NASTRAN performs its eigenvalue analysis. While this DMAP (and FORTRAN) should work with any version of NASTRAN 2001.0.1 and higher, it is recommended to use 2001.0.9 (r4).

Efi DMAP Input Summary

To perform an effective independence analysis, there are five decisions that the user must make. These are as follows

- Which specific EFI technique is desired?
- What are the candidate sensor degrees of freedom?
- What are the target modes?
- How many sensors are wanted in the final set?
- What printout options are desired?

The user controls the effective independence analysis by specifying the answers to these questions using a set of MSC/NASTRAN Bulk Data (PARAMs and DTI) and Case Control (SET and PARTN) entries. A summary of these entries is below.

Case Control Entries.

PARTN=n (integer)	Selects a set of grids for use as the initial "candidate sensor set".
-------------------	---

Bulk Data Parameters

TRIAX (YES/NO)	Selects tri-axial Efi algorithms
----------------	----------------------------------

As previously mentioned, the reduction of the initial set to the final set was a two-stage process. In the first stage, the candidate sensor set was reduced to 1000 nodes. The main features of the first stage analysis are listed below.

- Candidate set of approx. 19,000 nodes reduced to 1000 nodes
- 10 target modes
- Tri-axial Efl algorithm
- Mass weighting without Guyan redistribution
- DMAP with external FORTRAN program
- Run time of 7 hours 43 minutes on SGI Origin 3400 with "PARALLEL=4"

Note that the machine used to run the analysis was a shared memory multiprocessor (SMP) computer. Also, although the NASTRAN system cell "PARALLEL" was set to distribute the task across 4 processors, the Efl DMAP does not directly take advantage of multi-processing. However, many of the DMAP modules called by the Efl DMAP do. The primary benefit is seen in the initial and final calculations of the Efl program, where large matrices are reduced or large eigenvalue solutions are calculated. In the authors' experience, using multi-processing can increase performance, especially when Guyan reduction is used to reduce the NASTRAN ASET to the initial candidate sensor set.

The resulting set of grids from the first stage of the Efl analysis was then used as a starting set for the second stage. Several second stage analyses were performed. All of the second stage analyses used the more accurate mass weighting method in which a Guyan reduction is performed during each iteration to redistribute the mass to the remaining degrees of freedom. Each of the second stage analyses was identical with the exception of the size of the final set. Reductions were made to 100, 50, and 25 grids. These analysis sensor sets were turned over to the test engineers, who picked the final test sensor set from the Efl calculated sets based on considerations like symmetry, visualization, and accessibility of the sensor location.

The nodes calculated by the Efl analysis for the "100 node" set are shown in figure 2 as squares. Note that not all of the 100 nodes are visible in this picture. Table 1 contains a comparison of the target mode frequencies between the full model and the model reduced to the 100 grids of the calculated sensor set. As can be seen, the frequencies of table 1 show very good agreement between the full and the reduced model, with only one mode having a frequency error

over 5%. Table 2 is a cross orthogonality matrix between the mode shapes of the full model (columns) and the reduced model (rows). Table 2 shows that the target modes (which are the yellow, or shaded columns) can be easily identified from the 100 tri-axial accels.

For comparison purposes, a set of 100 grid points was selected using another common technique, average kinetic energy. The average KE was calculated using output from the GPKE (Grid Point KE) Case Control entry in MSC/NASTRAN. When this entry is specified, NASTRAN will print a table of the KE for each degree of freedom and for each calculated mode. Reference [4] lists the equation for this calculation as

$$KE = \Phi_g \otimes [M_{gg} \Phi_g]$$

Were the modes Φ_g are mass normalized and the operator $\otimes$ indicates term-wise matrix multiplication. A small FORTRAN program was then written to read the KE of the desired target modes, and calculate the average as

$$AveKE = \frac{1}{n} \sum_n \{KE_i\}$$

Here, $\{KE_i\}$ is the vector of kinetic energy for the i^{th} target mode, and n is the number of target modes. This formulation will yield the average kinetic energy for each degree of freedom. However, we are interested in the average kinetic energy for each grid, since we are comparing to a tri-axial Efl analysis that ranks grid importance, not degree of freedom importance. For this reason, the kinetic energies for each grid were taken to be the vector magnitude of the grids translational degrees of freedom. The KE magnitude for each grid was then averaged over the target modes. By sorting the resulting average KE vector, the 100 nodes with the highest average kinetic energy could be selected, and these nodes were used to generate NASTRAN ASET entries. The ASET entries were used with the MSC supplied "premaca" alter to calculate the cross orthogonality matrix shown in table 3. As can be seen from this table, the cross-orthogonality of the normal modes calculated from the reduced model using average KE is significantly poorer than that from the Efl program. In fact, the coupling terms in the cross-orthogonality matrix are large enough that it is difficult to judge from this matrix which mode from the full model corresponds to which mode of the reduced model.

Full Model (Hz)	Reduced Model (Hz)	% Difference
1.172	1.173	0%
1.219	1.221	0%
3.287	3.315	-1%
4.019	4.025	0%
4.439	4.517	-2%
4.648	4.712	-1%
5.599	5.744	-3%
6.819	7.044	-3%
8.620	9.189	-7%
10.107	10.600	-5%
12.970	13.239	-2%

Table 1. Comparison of frequencies from full model and 100 grid model calculate using Efl

HZ	1.172	1.219	3.287	4.019	4.439	4.648	5.599	6.819	8.620	10.107	10.822	10.828	10.858	11.198	11.277	11.397	11.906	12.094	12.867	12.970
1.173	-1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	-0.02	0.00	-0.01	0.00	0.00	-0.01	0.00	0.05	0.00	0.00	0.00
1.221	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	-0.01	0.00	-0.02	0.00	0.01	0.01	0.00	0.00	0.00	0.00	0.05	0.00
3.315	0.00	0.00	-1.00	-0.01	0.00	0.01	0.00	0.02	0.00	-0.03	0.00	0.00	0.00	0.00	-0.01	0.00	-0.03	0.00	0.00	0.00
4.025	0.00	0.00	0.00	-1.00	0.00	-0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.01	-0.01	0.01	0.00	-0.01	-0.01	0.00
4.517	0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.04	0.00	0.01	-0.01	-0.01	0.00	-0.01	-0.01	-0.01	-0.01	0.00	0.00
4.712	0.00	0.00	0.00	0.00	0.00	-1.00	0.00	0.01	0.00	-0.02	0.00	0.00	0.00	-0.02	-0.02	0.01	-0.03	0.00	0.00	0.00
5.744	0.00	0.00	0.00	0.00	0.00	0.00	-1.00	0.00	0.00	0.00	0.01	0.00	-0.03	0.01	0.00	0.03	0.00	-0.04	0.02	0.00
7.044	0.00	0.00	0.00	0.00	0.00	0.00	0.00	-1.00	0.00	-0.02	0.01	-0.02	0.00	-0.02	-0.03	0.00	0.00	0.01	0.01	0.00
9.189	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.01	-0.01	0.00	0.01	-0.01	0.01	-0.02	0.00	0.00	0.10	0.00
10.600	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	-0.01	1.00	0.00	-0.01	0.00	-0.02	-0.03	0.01	-0.10	0.00	-0.03	0.00
11.122	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	-0.01	-0.54	0.04	0.92	0.25	-0.33	0.05	-0.17	0.27	-0.12	0.00
11.572	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	-0.01	0.00	-0.10	-0.03	0.11	-0.68	0.28	-0.84	0.26	0.43	-0.02	0.00
12.772	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	-0.65	-0.26	-0.12	-0.35	0.35	0.33	-0.01	-0.29	0.06	0.00
13.042	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	-0.25	0.93	-0.12	0.00	0.07	-0.02	-0.18	-0.01	-0.01	-0.01
13.188	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.46	0.19	0.32	-0.33	0.33	0.25	-0.03	-0.37	0.04	0.00
13.239	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	-0.01	0.00	0.00	1.00
13.500	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	-0.02	0.03	-0.03	-0.05	-0.44	-0.68	0.10	-0.35	-0.04	-0.06	-0.03
13.672	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.02	-0.01	0.01	0.00	-0.04	-0.02	0.05	0.03	0.10	-0.06	-0.97	0.01
13.804	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	-0.08	0.00	-0.03	0.13	-0.19	-0.26	0.20	-0.67	0.04	0.01
14.442	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.01	-0.16	0.01	0.17	0.23	-0.15	-0.69	-0.09	-0.08	-0.04
15.009	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	-0.02	0.00	-0.04	0.00	0.01	0.02	-0.04	-0.27	-0.04	0.01	0.03

Table 2. Cross Orthogonality between full model and 100 grid model calculated using Efl

HZ	1.172	1.219	3.287	4.019	4.439	4.648	5.599	6.819	8.620	10.107	10.822	10.828	10.858	11.198	11.277	11.397	11.906	12.094	12.667	12.970
1.310	1.00	0.00	0.05	-0.04	0.06	0.63	-0.02	0.61	-0.01	-0.32	0.08	0.86	0.04	-0.28	0.43	-0.04	-0.67	0.02	0.03	-0.26
1.385	0.00	-1.00	0.00	0.00	-0.66	0.04	0.06	0.01	0.81	-0.01	0.10	0.35	-0.71	-0.27	-0.22	0.01	0.06	0.52	0.05	-0.02
3.810	0.00	0.00	-0.95	-0.40	0.01	0.14	-0.01	0.49	0.04	0.79	-0.02	0.07	0.03	-0.09	0.11	-0.01	0.09	0.00	0.02	-0.09
4.066	0.00	0.00	0.31	-0.92	-0.02	-0.20	0.00	-0.26	-0.02	-0.40	0.01	-0.04	-0.01	0.05	-0.06	0.01	-0.03	0.00	-0.01	0.05
6.247	0.00	0.00	0.00	0.00	-0.14	-0.02	-0.99	0.00	0.00	-0.18	-0.01	0.12	0.08	-0.01	-0.14	-0.05	0.16	-0.16	-0.02	
7.139	0.00	0.00	-0.01	0.02	-0.12	-0.72	0.03	0.19	0.00	0.21	-0.03	-0.19	0.01	-0.04	0.07	0.02	-0.50	-0.01	-0.07	0.54
7.448	0.00	0.00	0.00	0.00	-0.72	0.10	0.09	-0.04	-0.18	0.00	-0.20	0.02	0.13	0.16	0.19	-0.08	-0.14	-0.20	-0.80	-0.28
9.719	0.00	0.00	-0.02	0.01	0.01	-0.07	0.00	-0.53	0.02	-0.07	0.02	0.19	0.01	-0.08	0.09	0.00	0.07	-0.01	0.01	-0.29
11.002	0.00	0.00	0.00	0.00	0.00	0.00	-0.01	-0.01	-0.34	0.04	0.80	0.13	-0.62	0.57	0.75	-0.36	-0.40	-0.19	0.03	0.10
11.658	0.00	0.00	0.00	0.00	-0.01	0.00	-0.01	0.00	-0.12	0.00	0.36	-0.03	-0.24	-0.68	-0.29	0.91	0.29	-0.75	-0.13	-0.13
13.452	0.00	0.00	0.00	0.00	0.05	0.00	0.00	0.00	-0.41	0.00	-0.31	0.00	0.16	-0.07	-0.17	-0.07	0.00	0.18	-0.52	-0.18
15.602	0.00	0.00	0.00	0.00	0.00	0.00	0.00	-0.02	0.00	0.17	-0.01	0.06	0.01	-0.01	0.01	-0.01	-0.05	0.02	-0.01	0.01
18.296	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
20.544	0.00	0.00	0.00	0.00	0.00	-0.01	0.00	0.00	-0.02	-0.18	-0.02	-0.05	0.00	-0.02	0.06	-0.01	0.02	0.00	0.05	0.10
21.929	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	-0.01	-0.02	-0.01	0.01	0.00	-0.01	0.02	0.01	0.01	-0.02	0.04	0.08
22.469	0.00	0.00	0.00	0.00	0.00	0.01	0.00	-0.01	0.00	0.00	0.02	0.09	0.00	-0.04	0.08	-0.02	0.05	0.00	-0.03	0.61
23.073	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	-0.04	0.01	-0.07	0.03	0.01	-0.01	0.00	0.01	0.00	0.01	0.09	0.06
23.847	0.00	0.00	0.00	0.00	-0.01	0.00	0.00	0.00	-0.06	0.02	-0.13	0.05	0.01	0.01	0.01	0.02	0.02	0.01	0.17	0.06
24.709	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	-0.08	0.03	-0.02	-0.06	-0.13	-0.08	-0.02	-0.18	0.03	0.00
24.805	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	-0.01	0.00	0.00	0.00	0.00	0.00	0.01	0.00	0.01	0.03

Table 3. Cross Orthogonality and Frequency Errors for 100-grid reduction using average KE

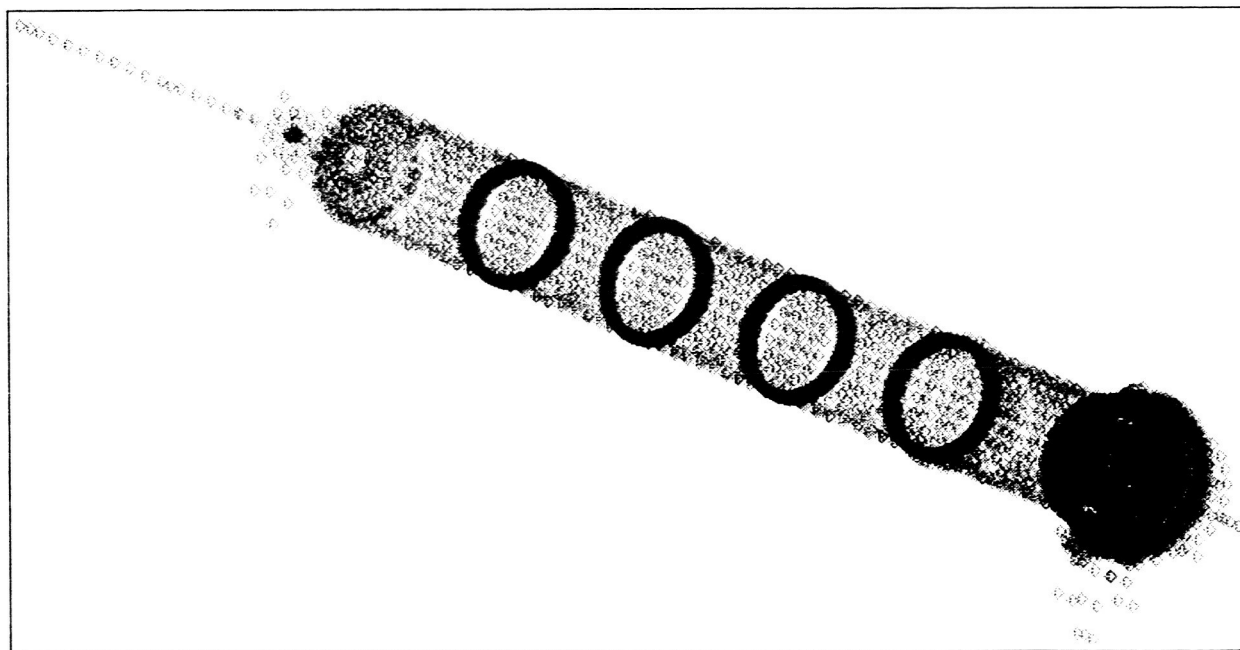


Figure 1. Initial Candidate Sensor Set

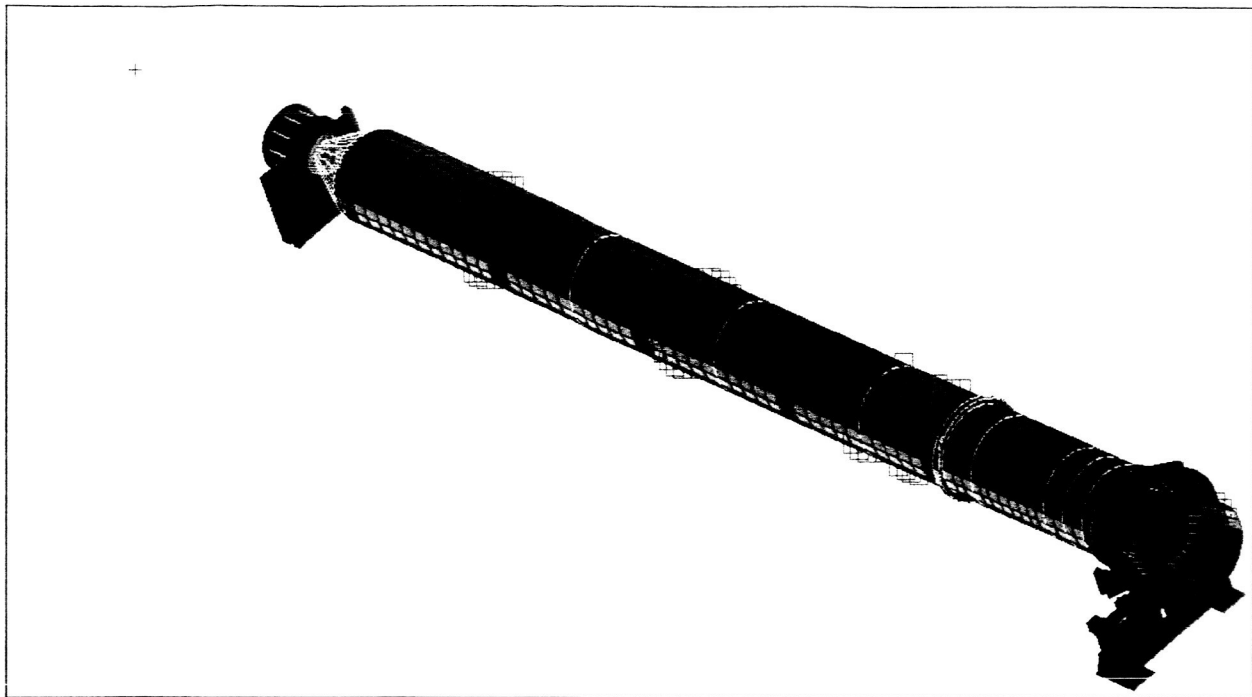


Figure 2, candidate sensor set reduced to 100 nodes (using EfI) shown on full model



Figure 3, candidate sensor set reduced to 100 nodes (using average KE) shown on full model